

An evolutionary algorithm for supply chain network design with assembly line balancing

Çağrı Koç¹

Received: 24 July 2015 / Accepted: 16 February 2016 / Published online: 2 March 2016
© The Natural Computing Applications Forum 2016

Abstract This paper investigates the combined impact of assembly line balancing decisions within a supply chain network design. The aim of the problem is to design a supply chain network between manufacturers, assemblers, and customers for specific periods, as well as balancing the assembly lines in assemblers. The main objective is to minimize the sum of transportation costs and fixed costs of stations in assemblers. Solving this problem poses several methodological challenges. To this end, the paper developed a powerful evolutionary algorithm (EA) which was successfully applied to a large pool of benchmark instances. The EA solved instances with up to 140 manufacturers and customers, and with up to 130 assemblers. Computational analyses are performed to empirically calculate the effect of various problem parameters, such as total cost, transportation cost and number of stations. The EA is validated on benchmark instances where it provides competitive solutions. Several managerial insights are also presented.

Keywords Supply chain network design · Assembly line balancing · Evolutionary metaheuristic

1 Introduction

A classical supply chain refers to a broad set of activities associated with the transformation and flow of goods, services and information from the main source to

customers (see [18]). Supply chain management includes the design of networks while determining positions of actors, optimizing amount of product flows between echelons and decreasing transportation costs. Network design process includes the location of facilities such as plants, distribution centers and retailers, design of the network configuration and satisfaction of customer demands (see [1, 11, 14, 32, 34]). There exists a rich scientific literature on the supply chain network design problem (SCND). Many mathematical models and heuristic algorithms have been developed to solve the SCND. These have gradually evolved from simple interchange schemes to more sophisticated metaheuristics, sometimes combining exact methods. Several survey articles have comprehensively reviewed and classified this rich literature from different perspectives. For a wealth of information on this topic, the reader is referred to the survey articles of [2, 19, 20, 24, 25, 28, 35] and the recent book of [36].

Assigning tasks to workstations while satisfying precedence relations between tasks and cycle time restrictions to optimize a performance measure is called as the assembly line balancing problem (ALB). This problem, in general, arises in the last stage of production processes. General performance of entire production systems is affected by the productivity level of assembly lines. The minimization of the number of workstation is generally considered as the performance measure. Three basic constraints of the ALB are as follows. The first one is the assignment constraints, i.e., each task must be assigned to one workstation. The second one is the precedence relations constraints, i.e., all precedence relations must be satisfied. The final one is the cycle time constraints, i.e., workstation workload must not exceed the cycle time. The workload is equal to the sum of the processing times of tasks in the related workstation. An assembly line consists of several successive workstations

✉ Çağrı Koç
cagri.koc@cirrelt.ca

¹ CIRRELT, Canada Research Chair in Distribution Management and HEC Montréal, Montréal H3T 2A7, Canada

where assembly operations are performed for a particular product. The smallest portion of an assembly operation is called as task. Completion time, i.e., cycle time, of tasks in the workstations constitutes the largest work content of assembly lines. Cycle time depends on the product demand during a planning horizon. The objective is to assign all given tasks with respect to precedence and cycle time constraints while minimizing the number of workstations required. The first known formulation of the ALB has been made by Salvesson [31]. This problem was later referred as the simple ALB by Baybars [7] which considers a single straight assembly line for only one type of product. Similar to the SCND, this problem has also been intensively studied in the literature. In recent years, many exact, heuristic and metaheuristic methods have been applied for solving the ALB (see [5]). For further details on the ALB and on its variants, the reader is referred to the survey articles of [6, 8, 9, 12, 15, 29, 33].

Gen et al. [13] studied the two-stage SCND and developed a priority-based genetic algorithm. The method includes several new decoding and encoding procedures, and a new crossover operator. Paksoy et al. [26] introduced the SCND with ALB (SCNDALB). The authors simultaneously consider both the SCND and the simple ALB first time in the literature. They formally defined the problem and proposed a mixed-integer nonlinear programming formulation to solve it. The objective minimizes the total cost of supply chain network and assembly operations. The authors conducted computational experiments on a small-size instance to demonstrate the effectiveness of the formulation. Paksoy and Özceylan [27] later introduced the SCND with U-type ALB. The study differs from [26] by considering U-type assembly line instead of straight one. The authors developed a nonlinear mixed-integer mathematical model and presented the results conducted on a small-size numerical example. Several scenarios were also created to analyze the problem and to provide several insights. Hamta et al. [16] considered the SCNDALB with demand uncertainty. The authors proposed a two-stage mixed-integer stochastic programming formulation which primarily based on two levels: strategic location decisions, as well as optimizing the SCNDALB with demand uncertainty. They combined a sample average approximation and several sampling procedures within the method. The results on new generated instances showed that the stochastic model yields better solutions than the deterministic model. The authors indicated that uncertainty is an important part of the formulation which can improve the quality of solutions. Yolmeh and Salehi [37] studied the SCNDALB under demand uncertainty. The authors developed a mixed-integer nonlinear programming formulation based on two-stage stochastic programming method and an outer approximation method. Their results

showed the efficiency of the proposed method. Hamta et al. [17] later proposed a bi-level programming model to solve the SCNDALB with push-pull strategy where decisions such as production amount and inventory level of each component in manufacturers are made. The authors first tested the performance of the method on a numerical example and then conducted computational experiments on several problem instances.

To our knowledge, this brief review shows that the SCNDALB has already been solved only by Paksoy et al. [26] on a small-size instance. We believe there exists merit for the development of a metaheuristic capable of efficiently solving the problem for larger-size instances within short computation times. This paper studies and analyzes the joint impact of assembly line balancing in a two-echelon supply chain. It makes three main scientific contributions. Our first contribution is to develop an evolutionary algorithm (EA). Our second contribution is to carry out computational experiments on generated instances ranges from small to large sizes. Our third contribution is to provide managerial insights in order to gain a deep understanding into the interactions between the components of the problem.

The remainder of this paper is structured as follows. Section 2 provides a general description of the problem and the mathematical formulation. Section 3 contains a detailed description of the metaheuristic. Computational experiments and analyses are presented in Sect. 4, followed by conclusions and managerial insights in Sect. 5.

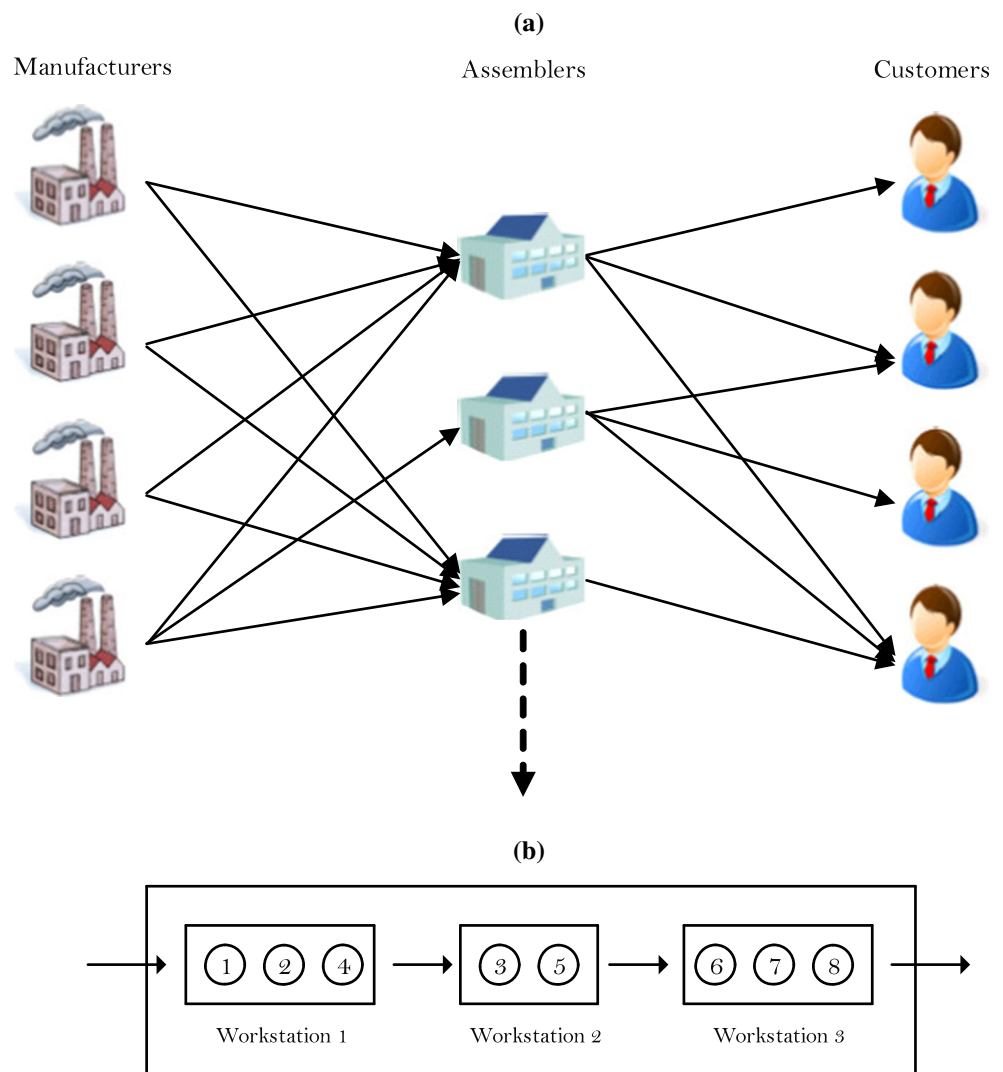
2 Problem definition and modeling

In the SCNDALB, one considers a multi-echelon supply chain with multiple manufacturers, components, assemblers, customers, multiple period time horizons and with a set of products to be manufactured or assembled, as well as assembly lines with stations. The problem consists of designing a network to minimize the total transportation costs between echelons for each period and the total fixed costs of stations in assemblers while determining the assignments of tasks to assembly lines. The supply chain network design and assembly line balancing problems are integrated through using cycle times which is the required time to complete a given process. Figure 1 illustrates the problem.

The main assumptions of the problem are as follows. The first four assumptions are commonly used for the SCND in the literature, while the others are the general assumptions of the ALB.

- Each customer has a positive demand which must be fully satisfied, i.e., shortages are not allowed.

Fig. 1 An illustration of the SCNDALB. **a** Supply chain network. **b** A straight and single assembly line



- Customer demands and transported materials are divisible such as in the case of gas or liquid products.
- The material flow is only allowed between two sequential echelons, i.e., from manufacturers to assemblers or from assemblers to customers.
- Each manufacturer and assembler have a limited capacity.
- Assembly lines produce one type of product which consists of several components.
- Each assembler has a variable cycle time.
- Operators travel times are considered zero.
- Work-in-process inventory is not allowed.
- A task cannot be split, and all tasks must be processed.
- The precedence relations of the problem are known.
- Any task can be processed at any station.
- Each process of the tasks has same costs.
- The task process time is independent of the station, and they are not sequence dependent.

- Serial lines are considered which have no feeder or parallel subassembly lines.
- Process times are additive at any station.
- Each model of a single product is processed on a specifically designed line.

2.1 Notations

We now present the indices, sets, parameters and decision variable of the formulation as follows.

Indices:

- m*: index of manufacturers.
- a*: index of assemblers.
- c*: index of customers.
- p*: index of periods.
- i, r, s*: index of tasks.
- j*: index of workstations.

Sets:

- M : number of manufacturers.
- A : number of assemblers.
- C : number of customers.
- P : number of periods.
- K : number of components.
- J : number of stations.
- N : number of tasks.
- L : set of tasks that precedes from a task.
- $(r, s) \in L$: a precedence relationship; r is an immediate predecessor of s .

Parameters:

- t_i : completion time of task i (time units).
- W_{time} : working time in period p (time units).
- a_{mkp} : capacity of manufacturer m for component k in period p (units).
- b_{ap} : capacity of assembler a in period p (units).
- u_{cp} : demand of customer c in period p (units).
- C_{map} : unit cost of shipping from manufacturer m to assembler a in period p (monetary units (MU)).
- C_{acp} : unit cost of shipping from assembler a to customer c in period p (MU).
- D_{ma} : distance between manufacturer m and assembler a (distance units).
- D_{ac} : distance between assembler a and customer c (distance units).
- O : fixed cost to open a station in the assembly line in all periods (MU).

Variables:

- X_{makp} : amount shipped from manufacturer m to assembler a for component k in period p .
- Y_{acp} : amount shipped from assembler a to customer c in period p .
- V_{aijp} : if task i is assigned to workstation j for assembler a in period p be equal to 1, and to 0 otherwise.
- Z_{ajp} if a task assigned to workstation j for assembler a in period p be equal to 1, and to 0 otherwise.
- CT_{ap} : cycle time for assembler a in period p .

2.2 Mathematical formulation

A nonlinear mixed-integer programming formulation of [26] for the SCNDALB is shown below:

$$\begin{aligned}
 \text{(SCNDALB) Minimize } & \sum_{m \in M} \sum_{a \in A} \sum_{k \in K} \sum_{p \in P} X_{\text{makp}} D_{\text{ma}} C_{\text{map}} \\
 & + \sum_{a \in A} \sum_{c \in C} \sum_{p \in P} Y_{\text{acp}} D_{\text{ac}} C_{\text{acp}} \tag{1}
 \end{aligned}$$

$$+ \sum_{a \in A} \sum_{j \in J} \sum_{p \in P} Z_{\text{ajp}} O \tag{2}$$

subject to

$$\sum_{a \in A} X_{\text{makp}} \leq a_{mkp} \quad m \in M, k \in K, p \in P \tag{3}$$

$$\sum_{c \in C} Y_{\text{acp}} \leq b_{ap} \quad a \in A, p \in P \tag{4}$$

$$\sum_{a \in A} Y_{\text{acp}} \geq u_{cp} \quad c \in C, p \in P \tag{5}$$

$$\sum_{m \in M} X_{\text{makp}} - \sum_{c \in C} Y_{\text{acp}} = 0 \quad a \in A, k \in K, p \in P \tag{6}$$

$$\sum_{j \in J} V_{\text{aijp}} = 1 \quad a \in A, i \in N, p \in P \tag{7}$$

$$\sum_{j \in J} V_{\text{arjp}} - \sum_{j \in J} V_{\text{asjp}} \leq 0 \quad a \in A, (r, s) \in L, p \in P \tag{8}$$

$$\sum_{i \in N} t_i V_{\text{aijp}} \leq CT_{\text{ap}} \quad a \in A, j \in J, p \in P \tag{9}$$

$$CT_{\text{ap}} = W_{\text{time}} / \sum_{c \in C} Y_{\text{acp}} \quad a \in A, p \in P \tag{10}$$

$$\sum_{i \in N} V_{\text{aijp}} - JZ_{\text{ajp}} \leq 0 \quad a \in A, j \in J, p \in P \tag{11}$$

$$X_{\text{makp}}, Y_{\text{acp}}, CT_{\text{ap}} \geq 0 \quad m \in M, a \in A, k \in K, c \in C, p \in P \tag{12}$$

$$V_{\text{aijp}}, Z_{\text{ajp}} \in \{0, 1\} \quad a \in A, i \in N, j \in J, p \in P. \tag{13}$$

The first term of the objective function (1) represents the total transportation costs of the first and second echelons for each period. The second term of the objective function (2) reflects the total fixed operating costs of stations in assemblers for each period.

Constraints (3) limit the total quantity of components shipped from manufacturers to assemblers. Constraints (4) enforce that the total quantity of product shipped from assemblers to customers cannot exceed the capacity of assemblers for each period. Constraints (5) satisfy the customer demand for all products for each period. Constraints (6) state that the total quantity of components shipped from manufacturers to assemblers must be equal to the total quantity of shipped products from assemblers to customers for each period. Constraints (7) ensure that each task is assigned to exactly one station in all assemblers for each period. Constraints (8) state the precedence relationship by assigning task r as an immediate predecessor of task s in all assemblers for each period. Constraints (9) guarantee that cycle time for a station cannot be exceeded for each period. Constraints (10) ensure that for all periods the cycle time is equal to the total working time which is divided by the total quantity of products. Constraints (11) state that if any task assigns to station j , it is in use for each assembler and for each period. Constraints (12) and (13) define the domains of the decision variables.

3 Description of the evolutionary algorithm

This section describes the EA to solve the SCNDALB. The EA builds on several powerful evolutionary-based meta-heuristic algorithms (see [3, 4, 10, 13, 21–23]).

The general structure of the EA is sketched in Algorithm 1. Initialization procedure generates the initial population (line 1). Two parents are selected (line 3) through a binary tournament process, and a crossover operation creates a new offspring C (line 4). The mutation procedure is applied to offspring C (Line 5). Each task is then assigned to assembly lines (line 6). Offspring C is then inserted into the population (line 7). As new offsprings are added, the population size n_a which is limited by $n_p + n_o$, changes over the iterations. The constant n_p denotes the size of the population initialized at the beginning of the algorithm, and the constant n_o denotes the maximum allowable number of offsprings that can be inserted into the population. If the population size n_a reaches $n_p + n_o$ at any iteration, then a survivor selection mechanism is applied (line 8). When the number of ϖ iterations without improvement in the incumbent solution is reached, the EA terminates (line 9).

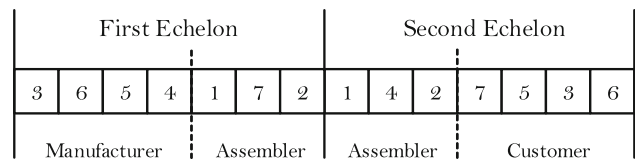


Fig. 2 An illustration of the priority-based encoding

represent the priority of each node. The length of this array is equal to the total number of echelons. We generate the transportation tree on a given solution by sequential arc appending between levels. Assemblers are first opened to satisfy customer demands where each tasks are assigned to stations. Manufacturers are then shipped required components to assemblers. The fitness value of each solution is calculated by using Eqs. (1) and (2). These fitness values are used to select survivors during the algorithmic iterations. An illustration of the priority-based encoding is given in Fig. 2. In priority-based encoding, each chromosome in the population consists of two parts. The first part represents transportation tree between manufacturers and assemblers,

Algorithm 1 The general framework of the EA

- 1: *Initialization*: Initialize a population with size n_p
 - 2: **while** the maximum number of iterations is reached = ϖ **do**
 - 3: *Parent selection*: select parent solutions P_1 and P_2
 - 4: *Crossover*: generate offspring C from P_1 and P_2
 - 5: *Mutation*: diversify the offspring C
 - 6: *Assembly line balancing*: assign the tasks of offspring C to assembly lines
 - 7: Add offspring C into the population
 - 8: *Survivor selection*: if the population size n_a reaches $n_p + n_o$, then select survivors
 - 9: **end while**
 - 10: Return best feasible solution
-

The remainder of this section introduces the main components of the EA. Representation and evaluation of the solution are given in Sect. 3.1. Section 3.2 presents the initialization of the population, the selections of parent solutions, a segment-based crossover operator and mutation procedure. Finally, the assembly line balancing procedure is described in Sect. 3.3.

3.1 Representation and evaluation

We adapted the priority-based encoding of [13] for the SCNDALB to represent our solutions within the population. A solution consists of priorities of first echelon including manufacturers and assemblers, and second echelon including assemblers and customers. A solution is a single-dimensional array, and numbers in there

and the second part represents transportation tree between assemblers and customers. In Fig. 2, we consider the problem that has four feasible manufacturers, three feasible assemblers and four customers. As it can be seen in Fig. 2, the first and second parts of chromosome consist of seven digits. We first decode the second part of chromosome since the total demand of customers has to be satisfied by assemblers. We simultaneously obtain transportation tree on this stage and decision related with which assemblers will be opened. We then obtain transportation tree on the first stage considering assemblers which were opened in the second stage. To satisfy the demand of assemblers, required numbers of manufacturers opened in this stage. For further implementation details on representation and evaluation, the reader is referred to [13].

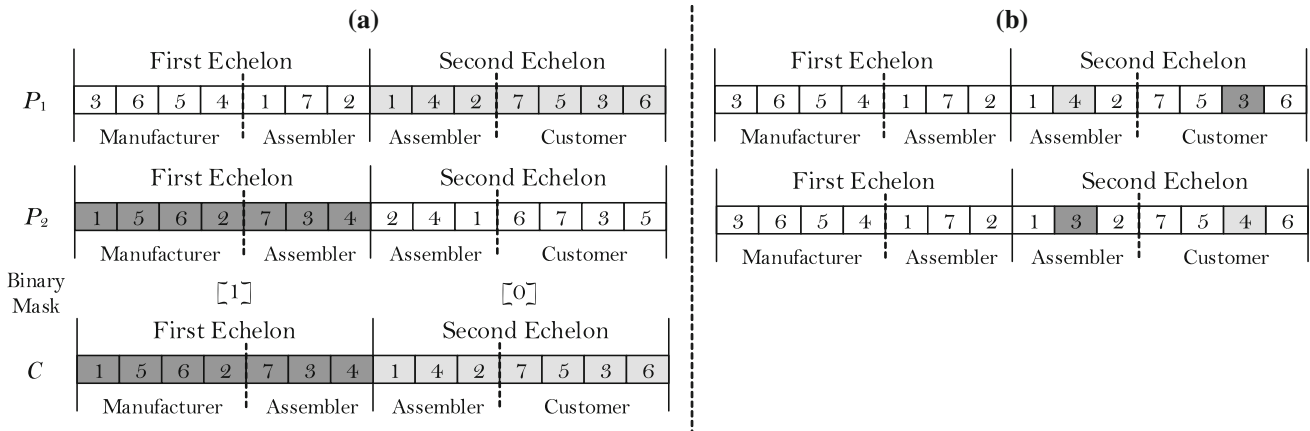


Fig. 3 An illustration of the crossover and mutation operators. **a** Crossover **b** Mutation

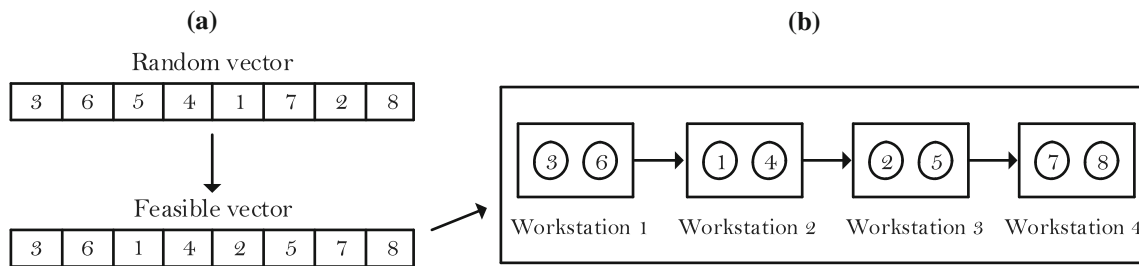


Fig. 4 An illustration of assembly line balancing. **a** Ordering procedure. **b** Division procedure

3.2 Genetic operators

The EA randomly generates the initial population. It randomly selects two parents from the population through a binary tournament to yield offspring *C* and keeps a parent having the best fitness value. Two parents then undergo to the segment-based crossover operator which aims to preserve good gene segments of both parents. Each echelon of offspring *C* is randomly selected with equal chance over echelons of parents. These crossover operators use a binary mask where its length is equal to the total number of echelons. To transfer the genetic materials from parents to offspring *C*, binary variables 0 and 1 are used, while 0 implies the first parent and 1 implies the second parent will transfer its genetic materials to the offspring *C*. An illustration of the segment-based crossover operator is given in Fig. 3a.

Population diversity plays a significant role in the efficient management of solutions. A segment-based mutation procedure is applied to offspring *C* where the priorities of two nodes are replaced to diversify the solution. The mutation operator first randomly selects an echelon using a binary mask as in the crossover operator, then randomly selects two nodes from this echelon, and finally exchanges their priorities with using swap method.

An illustration of the segment-based mutation operator is given in Fig. 3b.

The EA uses the survivor selection method (see [21–23]) to ensure the diversity of the population and to preserve the best solutions. The EA starts with an initial population of size n_p , and after each iteration an offspring is added to the population. When the current population size n_a reaches the maximum allowable size $n_p + n_o$, the survivor selection mechanism is put in place. This mechanism then selects n_p and discards n_o individuals from the population. The removal of n_o individuals is based on their fitness value where the best individuals are protected.

3.3 Assembly line balancing

When the cycle time for each period is determined, each task must be assigned to the stations in assemblers. We used a similar procedure to [30] to allocate tasks which includes two steps. In step 1 (Fig. 4a), we first randomly order each task and then transform it into a feasible solution according to a precedence relationship. In step 2 (Fig. 4b), we divide and assign the tasks to stations considering the cycle time.

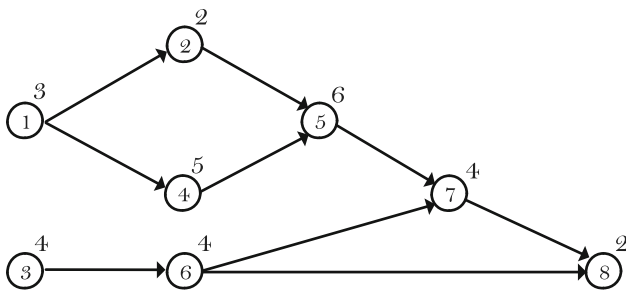


Fig. 5 Precedence diagram [26]

4 Computational experiments and analyses

This section presents the results of computational experiments. All computational experiments are conducted on a server with Intel Xeon 2.6 GHz processor. The EA is coded in C++.

4.1 Data sets and experimental settings

Evolutionary algorithms use a set of correlated parameters. We have conducted several experiments to further fine-tune these parameters. Following these tests, the following parameter values were used in our experiments: Since one offspring is generated by crossover and mutation operators at each generation, the crossover rate is set to 1, and the segment-based mutation is applied to offspring with the probability of 0.5. An initial population of size $n_p = 50$, and the maximum number of allowable offsprings in the population $n_o = 50$.

Benchmark data sets for the SCNDALB were generated by considering a similar procedure to [26]. In total, we generated 20 benchmark instances which consist of suppliers, assemblers and customers. In assemblers, we consider a single-model eight-task assembly line of [26] and the precedence diagram is shown in Fig. 5 where the task times range from two to six minutes.

Manufacturers produce seven different components and transport them to assemblers to produce end-products. When the assemble process is finished in assemblers, end-products are transported to customers. For each period, the distances between suppliers to assemblers and assemblers to customers, and unit transportation costs were randomly generated in the range [250, 600] and [0.1, 0.3], respectively. Components capacities of manufacturers for each period and for each component were randomly generated in the range [6000, 8000]. Capacities of assemblers and demand of customers were randomly generated in the range [4000, 7000] and [1200, 1600], respectively. Fixed cost to open a station in the assembly line in all periods is fixed to 100 (MU). Total number of

Table 1 Results of validation test on the base instance

	POG12	EA	Dev (%)
Opened stations	12	12	0.00
Transportation cost (MU)	40,65,360.40	40,65,360.40	0.00
Total cost (MU)	40,65,364.00	40,65,364.00	0.00
Time (s)	140.00	8.45	

periods, components, stations and tasks are two, seven, eight and eight, respectively.

4.2 Validation tests

We have conducted preliminary experiments on benchmark instances which are generated by Paksoy et al. [26] to assess the accuracy and the quality of the EA. To this end, we have compared the solution values of the EA with the optimal values obtained by Paksoy et al. [26]. The authors considered a single-model eight-task assembly line balancing problem (see Fig. 5) and the following data $M = 4, C = 4, P = 2, K = 7, J = 6, N = 8, O = 0.3$ MU, a period is 3 months or 12 weeks, and 5 working days in a week with 8 working hours in 1 day. In total, $W_{time} = 12 \times 5 \times 8 \times 60 = 28,000$ min.

We first solved the base instance of POG12 ([26]) and present its results in Table 1. These results clearly indicate that the EA yields optimal solution on this instance within short computation time.

We have conducted additional validation tests on Scenario A of POG12. In this scenario, the authors five times decreased and five times increased the customer demand ranges between 5 and 25 %. The results are presented in Table 2. As in the base case, the EA is able to find optimal solutions on most of the instances within short computation time. The results of validation tests clearly show that the EA is highly effective on the SCNDALB.

4.3 Results obtained on the test instances

This section presents the results obtained by the EA on 20 new generated benchmark instances. Table 3 presents the results for each instance where the columns display the number of manufacturers (|M|), number of assemblers (|A|), number of customers (|C|), total number of opened stations in periods 1 and 2, total transportation costs in periods 1 and 2, total cost and time (s). In these instances, for manufacturers, assemblers and customers, |M|, |A| and |C| range from four to 140, two to 130 and four to 140, respectively. Table 3 indicates that total number of opened stations, total transportation costs and total costs increases from instance 1 to instance 20. Solution times also follow similar magnitude which ranges from 1.97 seconds to 217.32 seconds.

Table 2 Results of validation tests on Scenario A of [26]

	POG12	EA	Dev (%)	POG12	EA	Dev (%)	POG12	EA	Dev (%)	POG12	EA	Dev (%)	POG12	EA	Dev (%)
Change in demand (%)	-25			-20			-15			-10			-5		
Opened stations	11	11	0.00	12	12	0.00	12	12	0.00	12	12	0.00	12	12	0.00
Total cost (MU)	30,49,019.70	30,49,019.70	0.00	32,52,288.40	32,52,288.40	0.00	34,90,356.40	34,90,356.40	0.00	36,58,824.40	36,58,824.40	0.00	38,62,092.40	38,62,092.40	0.00
Time (s)	11.00	10.00		10.00	9.00		11.00	10.00		41.00	28.00		62.00	30.00	
Change in demand (%)	+5			+10			+15			+20			+25		
Opened stations	13	13	0.00	14	14	0.00	14	14	0.00	15	15	0.00	16	16	0.00
Total cost (MU)	42,68,628.10	42,68,628.10	0.00	44,71,895.80	44,71,895.80	0.00	46,75,163.80	46,75,163.80	0.00	48,78,431.50	48,84,931.50	0.13	50,81,700.20	50,97,090.20	0.30
Time (s)	55.00	41.00		788.00	128.00		37.00	32.00		15576.00	256.00		29403.00	421.00	

4.4 Comparison of classical and integrated approaches

This section comparatively analyzes the classical approach in the literature and our integrated approach for solving the problem. In the classical approach, we first solve the SCND and yield a result about the amount shipped from a manufacturer to an assembler. We then determine the cycle time value of the assembly line in related assembler depending on this amount, i.e., the outputs of the SCND are used as inputs for the ALB. Integrated approach means the SCNDALB. We conduct experiments on five instances, SCNDALB-1, SCNDALB-2, SCNDALB-3, SCNDALB-4 and SCNDALB-5. Table 4 presents the results of this comparative analysis. It can be clearly seen from Table 4 that the SCNDALB, i.e., integrated approach, has cost advantage over the classical approach. On average, the total cost is decreased by -4.88 %. The computation times are similar magnitude.

4.5 The effect of variations in customer demand

In this section, we analyze the effect of variations in customer demand. To this end, we have conducted experiments using four different versions of the customer demand. In these tests, we decrease the customer demand by 30 and 15 %, and increase it by 15 and 30 %. The experiments were conducted on selected five instances SCNDALB-1, SCNDALB-5, SCNDALB-10, SCNDALB-15 and SCNDALB-20. Dev_{TC} (%) and Dev_{OS} (%) show the percentage deviation, between no-change case and change case, in total cost and in number of opened stations, respectively.

Looking at the results in Table 5, it is clear that total cost is decreased when the customer demand is decreased, and total cost increased when the customer demand increased. When we decrease the demand by 30 and 15 %, the average decrease in total cost is -28.38 and -14.80 %, respectively. When we increase the demand by 15 and 30 %, the average increases in total cost are 16.44 and 37.22 %, respectively. These results indicate that demand increase results in transportation of more products from manufacturers to customers via assemblers. Reversely, demand decrease results in transportation of less products to customers. We observe a similar situation in the total number of opened stations. Increase in demand decreases the cycle time in each assembler and results in opening new assembly stations for to satisfy this additional customer demand. When we increase the demand by 15 and 30 %, the average number of opened stations is increased by 6.96 and 12.80 %, respectively. When we decrease the demand by 30 and 15 %, the average opened stations decreased by -20.26 and

Table 3 Results on the benchmark instances

Instance	M	A	C	EA					
				Opened stations in period 1	Opened stations in period 2	Transportation cost in period 1	Transportation cost in period 2	Total cost	Time (s)
SCNDALB-1	4	2	4	4	6	24,22,946.25	17,27,137.83	41,50,584.08	1.97
SCNDALB-2	8	4	8	10	13	38,85,176.90	42,93,458.09	81,80,134.99	3.84
SCNDALB-3	10	5	10	18	9	65,97,169.41	51,00,652.48	11,69,9921.89	4.73
SCNDALB-4	15	10	15	43	46	88,07,235.71	65,69,943.77	15,38,5579.48	9.80
SCNDALB-5	20	15	20	54	62	90,32,299.33	10,33,5854.84	19,37,7854.17	16.43
SCNDALB-6	25	20	25	67	65	10,87,5407.81	11,12,7577.43	22,01,5485.24	17.22
SCNDALB-7	30	25	30	81	86	12,45,1294.90	13,64,5703.50	26,11,2498.40	21.57
SCNDALB-8	35	30	35	93	74	15,48,5708.81	16,43,8901.75	31,93,9510.56	31.37
SCNDALB-9	40	35	40	111	104	17,18,2788.13	18,03,7687.67	35,24,0675.80	38.09
SCNDALB-10	45	40	45	113	124	20,53,9814.01	19,68,1690.96	40,24,3004.97	36.12
SCNDALB-11	50	45	50	149	163	22,07,4999.92	23,31,0393.96	45,41,5693.88	41.64
SCNDALB-12	60	50	60	169	143	27,62,7825.00	26,21,9022.86	53,87,6647.86	46.51
SCNDALB-13	70	60	70	183	186	29,25,4669.20	29,81,8327.20	59,10,8196.40	59.09
SCNDALB-14	80	70	80	235	254	33,91,9170.30	35,53,1889.53	69,49,8059.83	72.34
SCNDALB-15	90	80	90	282	248	42,69,2605.61	40,22,1551.18	82,96,4756.79	78.71
SCNDALB-16	100	90	100	298	273	44,88,5534.49	43,16,5672.30	88,10,4506.79	92.68
SCNDALB-17	110	100	110	338	327	47,07,0133.22	48,47,4415.06	95,60,8648.28	133.70
SCNDALB-18	120	110	120	340	350	50,08,1490.70	47,45,3099.39	97,60,1190.09	132.61
SCNDALB-19	130	120	130	392	349	57,31,7583.46	51,94,4364.73	10,93,38248.19	202.23
SCNDALB-20	140	130	140	395	448	60,97,6722.93	60,28,4797.08	12,13,47120.01	217.32

Table 4 Results of comparative analysis

Instance	M	A	C	Classical approach		SCNDALB		Dev (%)
				Total cost	Time (s)	Total cost	Time (s)	
				SCNDALB-1	4	2	4	
SCNDALB-2	8	4	8	83,69,934.99	3.81	81,80,134.99	3.84	-2.32
SCNDALB-3	10	5	10	12,69,9921.89	4.68	11,69,9921.89	4.73	-8.55
SCNDALB-4	15	10	15	16,18,5579.48	9.85	15,38,5579.48	9.80	-5.20
SCNDALB-5	20	15	20	19,87,7854.17	17.45	19,37,7854.17	16.43	-2.58
Avg					7.56		7.35	-4.88

-8.11 %, respectively. Again, these results show that variations in customer demand directly affect the number of opened stations in assemblers and transportation costs between manufacturers, assemblers and customers.

4.6 The effect of variations in manufacturer capacity

We now investigate the effect of variations in manufacturer capacity. To do so, we have conducted experiments using four different versions of the manufacturer capacity. We decrease them by 30 and 15 %, and increase them by 15 and 30 %. As in Sect 4.5, the experiments were conducted on the same five instances.

Table 6 presents the results of this experiment. When we decreased manufacturers capacity by 30 %, on average the total cost is decreased by 2.07 %. However, on average the total cost is increased by 1.93 % when we decreased manufacturers capacity by 15 %. Similarly, on average the total cost increased by 1.50 and 4.20 %, when we increased manufacturers capacity by 15 and 30 %, respectively. These results indicate that change in the total cost ranges from -2.07 to 4.20 %, which is not as high as Table 5. We can conclude that variations in manufacturer capacity do not have the same effect on total cost as variations in customer demand. Our results also show that when we decrease the capacity by 30 and 15 %, on average the number of opened stations in assemblers are decreased by

Table 5 The effect of decreasing and increasing customer demand

Instance	M	A	C	Change in demand (%)	EA					
					Opened stations	Transportation cost	Total cost	Time (s)	Dev _{TC} (%)	Dev _{OS} (%)
SCNDALB-1	4	2	4	−30	8	28,83,853.37	28,84,653.37	1.72	−30.50	−20.00
SCNDALB-5	20	15	20	−30	90	14,34,8685.14	14,35,7685.14	12.07	−25.91	−22.41
SCNDALB-10	45	40	45	−30	190	29,86,0041.53	29,87,9041.53	28.54	−25.75	−19.83
SCNDALB-15	90	80	90	−30	401	60,10,2642.75	60,14,2742.75	69.1	−27.51	−24.34
SCNDALB-20	140	130	140	−30	719	82,15,7852.32	82,22,9752.32	177.65	−32.24	−14.71
Average					281.60			57.82	−28.38	−20.26
SCNDALB-1	4	2	4	−15	9	35,01,821.95	35,02,721.95	1.56	−15.61	−10.00
SCNDALB-5	20	15	20	−15	110	17,75,9542.21	17,77,0542.21	11.65	−8.29	−5.17
SCNDALB-10	45	40	45	−15	221	35,32,0514.35	35,34,2614.35	36.03	−12.18	−6.75
SCNDALB-15	90	80	90	−15	445	67,12,9117.83	67,17,3617.83	71.57	−19.03	−16.04
SCNDALB-20	140	130	140	−15	821	98,37,5181.44	98,45,7281.44	208.26	−18.86	−2.61
Average					321.20			65.81	−14.80	−8.11
SCNDALB-1	4	2	4	+15	11	49,06,757.11	49,07,857.11	2.45	18.24	10.00
SCNDALB-5	20	15	20	+15	126	23,09,0512.23	23,10,3112.23	12.29	19.22	8.62
SCNDALB-10	45	40	45	+15	249	46,60,3286.04	46,62,8186.04	33.95	15.87	5.06
SCNDALB-15	90	80	90	+15	560	91,02,1627.96	91,07,7627.96	80.6	9.78	5.66
SCNDALB-20	140	130	140	+15	889	14,44,12090.6	14,45,00990.6	183.14	19.08	5.46
Average					367.00			62.49	16.44	6.96
SCNDALB-1	4	2	4	+30	12	60,16,690.71	60,17,890.71	2.31	44.99	20.00
SCNDALB-5	20	15	20	+30	135	26,35,2532.21	26,36,6032.21	12.74	36.06	16.38
SCNDALB-10	45	40	45	+30	255	52,36,5579.56	52,39,1079.56	73.92	30.19	7.59
SCNDALB-15	90	80	90	+30	587	11,34,38961.1	11,34,97661.1	140.33	36.80	10.75
SCNDALB-20	140	130	140	+30	921	16,74,12090.6	16,75,04190.6	193.98	38.04	9.25
Average					382			84.66	37.22	12.80

6.19 and 1.97 %, respectively. On the other hand, when we increase the capacity by 15 and 30 %, on average opened stations are increased by 3.93 and 5.71 %, respectively. These results indicate that the number of opened stations in assemblers follows a similar pattern to the variations in manufacturer capacity.

4.7 The effect of variations in assembler capacity

This section analyzes the effect of variations in assembler capacity. To this end, we have conducted experiments on selected five instances using four different versions of the assembler capacity. We decrease their capacity by 30 and 15 %, and increase by 15 and 30 %.

Table 7 shows the effect of decreasing and increasing assembler capacity. When we look at the percentage deviation in the total cost, it is clear that we reach similar results as in Table 6. When we decrease the assemblers capacity by 30 %, on average the total cost is increased by 6.53 %. On the other hand, the total cost is decreased by 1.05, 0.43 and 2.12 %, when we decrease the capacity by

15 %, increase by 15 and 30 %, respectively. As in Table 6, variations in assembler capacity do not have the same effect on total cost as variations in customer demand. Table 7 shows that when we decrease the capacity by 30 %, on average number of opened stations in assemblers are decreased by 5.23 %. On the other hand, when we decrease the capacity by 15 % and increase by 15 and 30 %, on average opened stations are increased by 1.61, 9.53 and 11.37 %, respectively. These results indicate that the number of opened stations decrease, when we decrease the assembler capacity by 30 %. However, it does not follow the same pattern for 15 % decrease.

5 Conclusions and managerial insights

We have studied, analyzed and investigated the joint impact of assembly line balancing in a two-echelon supply chain network. We have developed an evolutionary algorithm (EA) to solve the problem. We have carried out computational experiments and analyses in order to gain a

Table 6 The effect of decreasing and increasing manufacturer capacity

Instance	M	A	C	Change in capacity (%)	EA					
					Opened stations	Transportation cost	Total cost	Time (s)	Dev _{TC} (%)	Dev _{OS} (%)
SCNDALB-1	4	2	4	-30	9	42,53,020.84	42,54,020.84	2.57	2.49	-10.00
SCNDALB-5	20	15	20	-30	110	19,06,0247.09	19,07,1047.09	13.79	-1.58	-5.17
SCNDALB-10	45	40	45	-30	220	38,25,7711.81	38,27,8911.8	38.99	-4.88	-7.17
SCNDALB-15	90	80	90	-30	509	77,91,0285.21	77,95,5185.21	109.58	-6.04	-3.96
SCNDALB-20	140	130	140	-30	804	12,08,77158.7	12,09,38258.7	210.24	-0.34	-4.63
Average					330.40			75.03	-2.07	-6.19
SCNDALB-1	4	2	4	-15	10	41,50,084.08	41,50,984.08	2.47	0.01	0.00
SCNDALB-5	20	15	20	-15	112	19,13,2940.85	19,14,2140.85	18.27	-1.22	-3.45
SCNDALB-10	45	40	45	-15	229	40,55,1714.65	40,57,3714.65	45.44	0.82	-3.38
SCNDALB-15	90	80	90	-15	521	85,31,4494.65	85,36,3694.66	109.99	2.89	-1.70
SCNDALB-20	140	130	140	-15	832	12,99,36628.5	13,00,11428.5	210.55	7.14	-1.30
Average					340.80			77.34	1.93	-1.97
SCNDALB-1	4	2	4	+15	11	41,50,084.08	41,50,984.08	2.35	0.01	10.00
SCNDALB-5	20	15	20	+15	119	19,61,6259	19,62,6059	21.55	1.28	2.59
SCNDALB-10	45	40	45	+15	241	41,09,5158.14	41,11,6358.15	53.33	2.17	1.69
SCNDALB-15	90	80	90	+15	539	81,52,7638.12	81,57,6538.11	126.04	-1.67	1.70
SCNDALB-20	140	130	140	+15	874	12,81,98950.9	12,82,80050.9	176.93	5.71	3.68
Average					356.80			76.04	1.50	3.93
SCNDALB-1	4	2	4	+30	11	41,50,084.08	41,50,584.08	2.45	0.00	10.00
SCNDALB-5	20	15	20	+30	123	20,20,0635.55	20,21,0035.56	19.27	4.29	6.03
SCNDALB-10	45	40	45	+30	248	42,43,0161.42	42,44,9461.42	56.79	5.48	4.64
SCNDALB-15	90	80	90	+30	544	82,04,2093.97	82,09,1893.96	117.28	-1.05	2.64
SCNDALB-20	140	130	140	+30	887	13,61,63755.7	13,62,44455.7	236.98	12.28	5.22
Average					362.6			86.55	4.20	5.71

deep understanding into the interactions between the components of the problem. We have derived managerial insights by investigating the effect of various problem components on total cost and number of opened stations in assemblers. In what follows, we summarize our main conclusions.

Our algorithm is able to solve small- to large-size instances, with up to 140 manufacturers and customers, and with up to 130 assemblers, within reasonable computation times. The EA yields the highly competitive solutions on benchmark instances of [26] within short computation time. Our comparison between the classical and integrated approaches clearly shows that the SCNDALB has cost advantage over the classical approach. The analyses quantify the benefits of using the integrated approach.

In supply chain network, as expected, our experimental results indicate that customer demand increase results in transportation of more products from manufacturers to customers via assemblers and customer demand decrease results in transportation of less products to customers. We have also shown that variations in manufacturer capacity

and in assembler capacity do not have the same effect on total cost as variations in customer demand.

We have demonstrated that increase in customer demand decreases the cycle time in each assembler and results in opening new assembly stations for to satisfy this customer demand. Our results shows that the number of opened stations in assemblers follows a similar pattern to the variations in manufacturer capacity. However, when we decrease the assembler capacity by 30 %, the number of opened stations decreases, but it does not follow the same pattern for 15 % decrease. These analyzes show that variations in customer demand, in assembler capacity or in manufacturer capacity, directly affect the number of opened stations in assemblers and transportation costs between manufacturers, assemblers and customers.

Our results depend on the parameter values used in the experimental design. Beyond our computational experiments, we indicate the importance of the availability of our method which is capable of analyzing the trade-offs between assembly line balancing and two-echelon supply

Table 7 The effect of decreasing and increasing assembler capacity

Instance	M	A	C	Change in capacity (%)	EA					
					Opened stations	Transportation cost	Total cost	Time (s)	Dev _{TC} (%)	Dev _{OS} (%)
SCNDALB-1	4	2	4	−30	9	45,73,223.82	45,74,923.82	2.75	10.22	−10.00
SCNDALB-5	20	15	20	−30	112	19,32,5437.65	19,34,0237.65	13.83	−0.19	−3.45
SCNDALB-10	45	40	45	−30	225	43,64,9556.68	43,68,2356.68	40.78	8.55	−5.06
SCNDALB-15	90	80	90	−30	511	86,81,1424.22	86,88,2624.23	102.81	4.72	−3.58
SCNDALB-20	140	130	140	−30	809	13,25,72881.6	13,26,86981.6	221.85	9.34	−4.03
Average					333.20			76.40	6.53	−5.23
SCNDALB-1	4	2	4	−15	11	42,87,550.95	42,89,250.95	2.43	3.34	10.00
SCNDALB-5	20	15	20	−15	116	18,43,6764.83	18,44,8464.83	13.18	−4.80	0.00
SCNDALB-10	45	40	45	−15	232	39,72,9734.25	39,75,5334.25	35.77	−1.21	−2.11
SCNDALB-15	90	80	90	−15	536	81,77,9391.81	81,83,1591.81	118.51	−1.37	1.13
SCNDALB-20	140	130	140	−15	835	11,97,94370	11,98,85170	163.32	−1.20	−0.95
Average					346.00			66.64	−1.05	1.61
SCNDALB-1	4	2	4	+15	12	41,19,790.53	41,20,390.53	1.63	−0.73	20.00
SCNDALB-5	20	15	20	+15	136	19,62,9445.98	19,63,6645.98	11.84	1.34	17.24
SCNDALB-10	45	40	45	+15	245	40,58,7266.81	40,60,1766.81	31.22	0.89	3.38
SCNDALB-15	90	80	90	+15	541	77,67,1962.84	77,69,9262.84	78.84	−6.35	2.08
SCNDALB-20	140	130	140	+15	885	12,45,56875.3	12,46,00775.3	177.88	2.68	4.98
Average					363.80			60.28	−0.43	9.53
SCNDALB-1	4	2	4	+30	12	43,29,962.94	43,31,062.94	2.04	4.35	20.00
SCNDALB-5	20	15	20	+30	139	17,85,4661.52	17,86,0161.52	9.72	−7.83	19.83
SCNDALB-10	45	40	45	+30	254	39,35,3021.36	39,36,2621.36	28.56	−2.19	7.17
SCNDALB-15	90	80	90	+30	552	79,04,7976.97	79,06,6876.96	78.01	−4.70	4.15
SCNDALB-20	140	130	140	+30	891	12,10,07683.4	12,10,45483.4	164.13	−0.25	5.69
Average					369.6			56.49	−2.12	11.37

chain network design. Further studies should focus on to improve the assembly line balancing process of the meta-heuristic algorithm by using new and effective heuristic procedures to gain some cost benefits.

Acknowledgments Thanks are due to the referees for their valuable comments.

References

1. Aburto L, Weber R (2007) Improved supply chain management based on hybrid demand forecasts. *Appl Soft Comput* 7:136–144
2. Akyuz GA, Erkan TE (2010) Supply chain performance measurement: a literature review. *Int J Prod Res* 48:5137–5155
3. Altiparmak F, Gen M, Lin L, Paksoy T (2006) A genetic algorithm for multi-objective optimization of supply chain networks. *Comput Ind Eng* 51:197–216
4. Altiparmak F, Gen M, Lin L, Karaoglan I (2009) A steady-state genetic algorithm for multi-product supply chain network design. *Comput Ind Eng* 56:521–537
5. Atasagun Y, Kara Y (2014) Bacterial foraging optimization algorithm for assembly line balancing. *Neural Comput Appl* 25:237–250
6. Battaia O, Dolgui A (2013) A taxonomy of line balancing problems and their solution approaches. *Int J Prod Econ* 142:259–277
7. Baybars İ (1986) A survey of exact algorithms for the simple line balancing problem. *Manag Sci* 32:909–932
8. Becker C, Scholl A (2006) A survey on problems and methods in generalized assembly line balancing. *Eur J Oper Res* 168:694–715
9. Boysen N, Flidner M, Scholl A (2007) A classification of assembly line balancing problems. *Eur J Oper Res* 183:674–693
10. Demirel N, Özceylan E, Paksoy T, Gökçen H (2014) A genetic algorithm approach for optimising a closed-loop supply chain network with crisp and fuzzy objectives. *Int J Prod Res* 52:3637–3664
11. Deshpande U, Gupta A, Basu A (2004) Task assignment with imprecise information for real-time operation in a supply chain. *Appl Soft Comput* 5:101–117
12. Gagnon R, Ghosh S (1991) Assembly line research: historical roots, research life cycles and future directions. *OMEGA* 19:381–399
13. Gen M, Altiparmak F, Lin L (2006) A genetic algorithm for two-stage transportation problem using priority-based encoding. *OR Spectr* 28:337–354
14. Ghiani G, Laporte G, Musmanno R (2013) Introduction to logistics systems management. Wiley, Chichester

15. Ghosh S, Gagnon R (1989) A comprehensive literature review and analysis of the design, balancing and scheduling of assembly lines. *Int J Prod Res* 27:637–670
16. Hamta N, Shirazi MA, Fatemi Ghomi SMTF, Behdad S (2014) Supply chain network optimization considering assembly line balancing and demand uncertainty. *Int J Prod Res* 53:2970–2994
17. Hamta N, Shirazi MA, Ghomi SF (2015) A bi-level programming model for supply chain network optimization with assembly line balancing and pushpull strategy. In: Proceedings of the institution of mechanical engineers, Part B: *J Eng Manuf*, 1–17
18. Kesen SE (2014) Capacity-constrained supplier selection model with lost sales under stochastic demand behaviour. *Neural Comput Appl* 24:347–356
19. Klibi W, Martel A, Guitouni A (2010) The design of robust value-creating supply chain networks: a critical review. *Eur J Oper Res* 203:283–293
20. Ko M, Tiwari A, Mehnen J (2010) A review of soft computing applications in supply chain management. *Appl Soft Comput* 10:661–674
21. Koç Ç, Bektaş T, Jabali O, Laporte G (2014) The fleet size and mix pollution-routing problem. *Trans Res Part B Methodol* 70:239–254
22. Koç Ç, Bektaş T, Jabali O, Laporte G (2015) A hybrid evolutionary algorithm for heterogeneous fleet vehicle routing problems. *Comput Oper Res* 64:11–27
23. Koç Ç, Bektaş T, Jabali O, Laporte G (2016) The fleet size and mix location-routing problem with time windows: formulations and a heuristic algorithm. *Eur J Oper Res* 248:33–51
24. Meixell MJ, Gargeya VB (2005) Global supply chain design: a literature review and critique. *Trans Res Part E Logis Trans Rev* 41:531–550
25. Melo MT, Nickel S, Saldanha-da-Gama F (2009) Facility location and supply chain management: a review. *Eur J Oper Res* 196:401–412
26. Paksoy T, Özceylan E, Gökçen H (2012) Supply chain optimisation with assembly line balancing. *Int J Prod Res* 50:3115–3136
27. Paksoy T, Özceylan E (2012) Supply chain optimisation with U-type assembly line balancing. *Int J Prod Res* 50:5085–5105
28. Pishvaei MS, Zanjirani Farahani R, Dullaert W (2010) A memetic algorithm for bi-objective integrated forward/reverse logistics network design. *Comput Oper Res* 37:1100–1112
29. Rekiek B, Dolgui A, Delchambre A, Bratcu A (2002) State of art of assembly lines design optimisation. *Annu Rev Control* 26:163–174
30. Rubinovitz J, Levitin G (1995) Genetic algorithm for assembly line balancing. *Int J Prod Econ* 41:343–354
31. Salveson M (1955) The assembly line balancing problem. *J Ind Eng* 6:18–25
32. Simchi-Levi D, Kaminsky P, Simchi-Levi E (2004) Managing the supply chain: the definitive guide for the business professional. McGraw-Hill, New York
33. Scholl A, Becker C (2006) State-of-the-art exact and heuristic solution procedures for simple assembly line balancing. *Eur J Oper Res* 168:666–693
34. Smirnov AV, Sheremetov LB, Chilov N, Cortes JR (2004) Soft-computing technologies for configuration of cooperative supply chain. *Appl Soft Comput* 4:87–107
35. Stadtler H (2005) Supply chain management and advanced planning-basics, overview and challenges. *Eur J Oper Res* 163:575–588
36. Stadtler H, Kilger C, Meyr H (eds) (2015) Supply chain management and advanced planning. Springer, Berlin
37. Yolmeh A, Salehi N (2015) An outer approximation method for an integration of supply chain network designing and assembly line balancing under uncertainty. *Comput Ind Eng* 83:297–306